

# **BeerTag**

# Drink beer socially

# Project Description

Your task is to develop **BeerTag** web application. **BeerTag** enables your users to manage all the beers that they have drunk and want to drink. Each beer has detailed information about it from the ABV (alcohol by volume) to the style and description. Data is community driven, and every beer lover can add new beers. Also, **BeerTag** allows you to rate a beer and see average ratings from different users. You can refer to <https://untappd.com/> as a real-world example.

# Functional Requirements

## Entities

#### Each **user** must have a name, email, and profile picture.

#### Name must be between 2 and 20 symbols.

* + Email must be valid email and unique in the system.

#### Each **beer** must have a name, description, style, ABV (Alcohol by Volume), brewery that produces it, and a picture.

#### Beer name must be unique and have between 2 and 20 symbols.

#### Each **style** must have a name.

#### Style name must be unique and between 2 and 20 symbols.

#### Each **brewery** must have a name and country.

#### Brewery name must be unique and have between 2 and 20 symbols.

#### Each **country** must have a name.

## Public Part

The public part must be accessible without authentication i.e., for anonymous users.

Anonymous users must be able to browse all beers and their details. They must be able to filter by name, ABV, country and style, and sort by name, ABV and rating.

Also, anonymous users must have the ability to register and login.

## Private part

Accessible only if the user is authenticated.

Registered users must be able to create new beers, add a beer to their wish list and drunk list and rate beers. They must be able to edit and delete their own beers. Registered users should be able to modify their personal information as well.

## Administrative part

Available to administrators only.

Administrators must be able to edit/delete all beers and users. Administrators should be able to modify breweries, and styles.

## REST API

To provide other developers with your service, you need to develop a REST API. It should leverage HTTP as a transport protocol and clear text JSON for the request and response payloads.

A great API is nothing without great documentation. The documentation holds the information that is required to successfully consume and integrate with an API. You must use [Swagger](https://swagger.io/) to document yours.

The REST API provides the following capabilities:

1. Countries
   * Read operations (must)
   * Create, Update, Delete operations (should)
2. Breweries
   * Read operations (must)
   * Create, Update, Delete operations (should)
3. Styles
   * Read operations (must)
   * Create, Update, Delete operations (should)
4. Beers
   * CRUD operations (must)
   * Filter by name, ABV, country and style (must)
   * Sort by name, ABV, rating (must)
   * Rate beer (must)
5. Users
   * CRUD operations (must)
   * Add beer to wish list (must)
   * Get wish list beers (must)
   * Add beer to drunk list (must)
   * Get drunk list beers (must)